CS4044D – MACHINE LEARNING

Course Project

Face Recognition: Attendance Tracking System

Department of Computer Science and Engineering
National Institute of Technology, Calicut

HARINARAYANAN J (B200741CS)

JACKSON STEPHAN (B200743CS)

Problem Definition

Deep learning has revolutionized the field of facial recognition due to its ability to automatically and adaptively learn spatial hierarchies of features from image data. Utilizing this technique to design a facial recognition model that can efficiently differentiate and correctly identify different faces in an image to help solve a real-life problem such as class attendance.

Proposed Methodology

1. Collection of Dataset:

For the purpose of creating the training database, around 30 photos per subject are obtained. For the scope of this project, we are using 5 subjects. The test dataset comprises of 16 unseen instances of the 5 subjects.

2. Preprocessing

Image is read in grayscale and transformed into a tensor. Labels are transformed using LabelEncoder. This technique converts each unique string value into a numerical value. Each class is assigned an integer value from 0 to N-1, where N is the number of unique classes.

3. Data Augmentation

Since the train data consists of facial images which involves a lot of features to focus on, it is important that there should be enough data available to avoid overfitting. So, in order for the model to generalize properly we had to apply some augmentations to the training data. The following are the augmentations used: Horizontal flips, Random crops, Linear contrast Multiply, Affine transformations, Elastic transformations.

4. Features extraction

The feature extraction process involves applying a series of convolutional, activation, normalization, pooling, and dropout operations to the input image, resulting in a set of learned feature

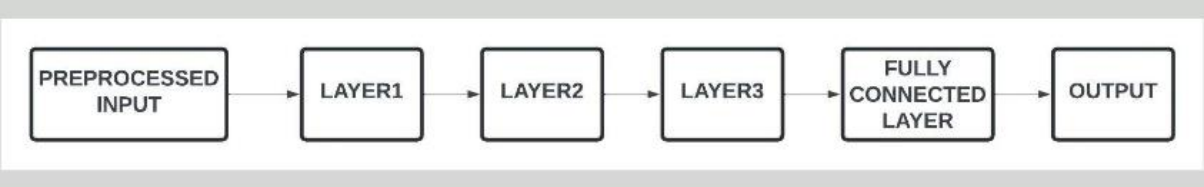
maps. These feature maps are then flattened and passed through a fully connected layer to make the final predictions.

5. Classification

The trained model is applied to the pre-processed image. A vector of probabilities, one for each class, is the model's output. This is accomplished by ensuring that the output probabilities add up to one by applying a Softmax function to the output of the final layer of the model. The class in the output vector with the highest probability is the predicted label for the image. The argmax function, which yields the index of the highest value in a tensor, is usually used to accomplish this. The assigned photos and output images are saved in a folder together. A CSV file is generated using the labels to track attendance.

Architecture Used

Overview Diagram



Layer1

Conv2d-1	[-1, 64, 200, 200]	640
BatchNorm2d-2	[-1, 64, 200, 200]	128
ReLU-3	[-1, 64, 200, 200]	0
Conv2d-4	[-1, 64, 200, 200]	36,928
BatchNorm2d-5	[-1, 64, 200, 200]	128
ReLU-6	[-1, 64, 200, 200]	0
Conv2d-7	[-1, 64, 200, 200]	102,464
BatchNorm2d-8	[-1, 64, 200, 200]	128
ReLU-9	[-1, 64, 200, 200]	0
MaxPool2d-10	[-1, 64, 100, 100]	0
Dropout2d-11	[-1, 64, 100, 100]	0

Layer2

Conv2d-12	[-1, 128, 100, 100]	73,856
BatchNorm2d-13	[-1, 128, 100, 100]	256
ReLU-14	[-1, 128, 100, 100]	0
Conv2d-15	[-1, 128, 100, 100]	147,584
BatchNorm2d-16	[-1, 128, 100, 100]	256
ReLU-17	[-1, 128, 100, 100]	0
Conv2d-18	[-1, 128, 100, 100]	409,728
BatchNorm2d-19	[-1, 128, 100, 100]	256
ReLU-20	[-1, 128, 100, 100]	0
MaxPool2d-21	[-1, 128, 50, 50]	0
Dropout2d-22	[-1, 128, 50, 50]	0

Layer3

Conv2d-23	[-1, 256, 50, 50]	295,168
BatchNorm2d-24	[-1, 256, 50, 50]	512
ReLU-25	[-1, 256, 50, 50]	0
MaxPool2d-26	[-1, 256, 25, 25]	0
Dropout2d-27	[-1, 256, 25, 25]	0

Fully connected layer

Linear-28	[-1, 256]	40,960,256
BatchNorm1d-29	[-1, 256]	512
ReLU-30	[-1, 256]	0
Linear-31	[-1, 128]	32,896
BatchNorm1d-32	[-1, 128]	256
ReLU-33	[-1, 128]	0
Linear-34	[-1, 5]	645
Softmax-35	[-1, 5]	0

Evaluation and Results

The model was run on a test set of size 16 with different instances of the 5 subjects and gave 87.5 percent of success rate

Input test images

Output images with assigned label